Underwater Vehicle Localisation using Extended Kalman Filter

Miroslav Radojević

Heriot-Watt University, Edinburgh
Universitat de Girona
Universitè de Bourgogne

miroslav.radojevic@gmail.com

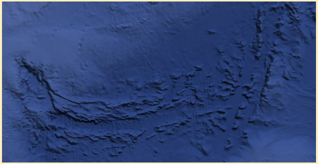
15th June 2011.

Motivation

Underwater?

Edinburgh, city centre.
$55°57'16''N3°11'58''W$
Elevation: $66m$
Area: $\approx 400 \times 300m$

... 1300 km away ...



Norwegian sea.
$66°52'46''N3°36'21''W$
Elevation: $-3335m$
Area: $\approx 700 \times 300km$

Why navigation?

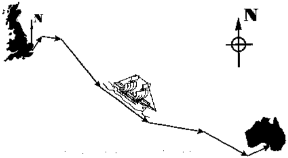
To be able to navigate the robot within the environment - we need to know it's position - to **localise** it.

Navigation

Navigation implies the capabilities of:

• accurate determination of the vehicle position and velocity with respect to a known reference point

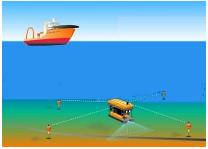
• planning and the execution of the movements between locations



John Harrison, 18^{th} century

longitude problem

GPS, USDOD, 1974-1994.



Underwater navigation strategies

✓ Measured depth is fairly accurate → localisation mainly 2d

✗ Obtaining absolute position is more complex underwater

✗ Cameras and sonar are of a limited use

✗ Dead-reckoning sensors are expensive and prone to drifting

⟹ use all available inertial information to navigate

Autonomous Underwater Vehicle (AUV)

Sensors

- Pressure sensor: depth

✓ precise absolute depth

- Digital compass: 3d attitude

✗ magnetic declination ✗ prone to disturbances

✓ absolute measurement

- Fibre-optic gyro: orientation rate

✓ high accuracy

- Doppler Velocity Log (DVL): 3d speed vector, altitude

✗ needs certain altitude to work

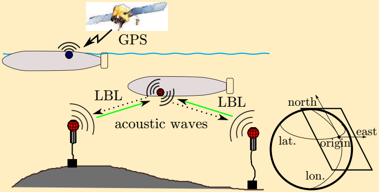
- GPS: latitude, longitude

✗ only on the surface

- Long-baseline (LBL): absolute 3d position - north, east, depth

Platform: Nessie AUV

Absolute positioning

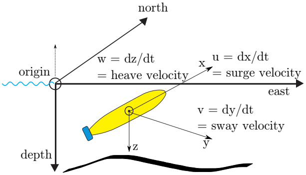


Vehicle Navigation State Vector

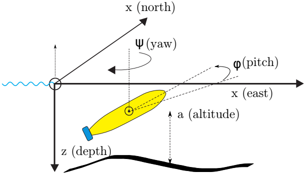
Vehicle state is a vector that contains variables relevant for localising the vehicle (Eq. 1). Vehicle navigation state describes its position and motion within the environment. Elements of the state vector $\mathbf{X}(k)$ are treated as Gaussian Random Variables (GRV). State vector will combine angular and metric values.

$$\mathbf{X}(k) = \begin{bmatrix} x & y & z & a & u & v & w & \psi & \varphi & \dot{\psi} & \dot{\varphi} \end{bmatrix}^T \quad (1)$$

$\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{a}$: *north*, *east*, *depth* and *altitude* (Fig. 1(b)).
$\mathbf{u}, \mathbf{v}, \mathbf{w}$: linear velocities w.r.t sea bed in vehicle's coordinate system (*surge*, *sway* and *heave*, Fig. 1(a))
$\boldsymbol{\psi}, \boldsymbol{\varphi}$: *yaw* and *pitch* (vehicle orientation, Fig. 1(b)).
$\dot{\boldsymbol{\psi}}, \dot{\boldsymbol{\varphi}}$: *yaw rate* and *pitch rate* (angular velocities).



(a) velocities w.r.t. sea bed



(b) global positioning

Figure 1: AUV positioning

5 d.o.f. system model is describing how the state $\mathbf{X}(k)$ evolves in time. It is a *constant speed* model [1] that uses previous state $\mathbf{X}(k-1)$ corrupted with *zero-mean* GRV with linear and angular acceleration noise $\mathbf{N}(k-1)$ to make a prediction of the next state vector value (Fig. 2, Eq. 3, 2).

$$\mathbf{X}(k) = f(\mathbf{X}(k-1), \mathbf{N}(k-1)) \tag{2}$$
$$\mathbf{N}(k) = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} & \ddot{\psi} & \ddot{\varphi} \end{bmatrix}^T$$

$$\begin{bmatrix} x \\ y \\ z \\ a \\ \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \\ \psi \\ \varphi \\ \dot{\boldsymbol{\psi}} \\ \dot{\boldsymbol{\varphi}} \end{bmatrix}_{(k)} = \begin{bmatrix} x + (uT + \dot{u}\frac{T^2}{2})\cos(\psi)\cos(\varphi) - (vT + \dot{v}\frac{T^2}{2})\sin(\psi)\cos(\varphi) \\ y + (uT + \dot{u}\frac{T^2}{2})\sin(\psi)\cos(\varphi) + (vT + \dot{v}\frac{T^2}{2})\cos(\psi)\cos(\varphi) \\ z + (wT + \dot{w}\frac{T^2}{2})\cos(\varphi) \\ a - (wT + \dot{w}\frac{T^2}{2})\cos(\varphi) \\ \mathbf{u} + \dot{u}T \\ \mathbf{v} + \dot{v}T \\ \mathbf{w} + \dot{w}T \\ \psi + \dot{\psi}T + \ddot{\psi}\frac{T^2}{2} \\ \varphi + \dot{\varphi}T + \ddot{\varphi}\frac{T^2}{2} \\ \dot{\boldsymbol{\psi}} + \ddot{\psi}T \\ \dot{\boldsymbol{\varphi}} + \ddot{\varphi}T \end{bmatrix}_{(k-1)} \tag{3}$$

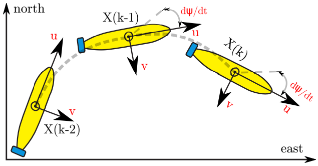
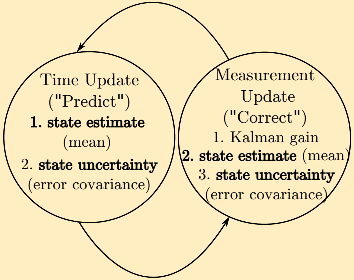


Figure 2: State transition model

Extended Kalman Filter (EKF)

The goal is to estimate the vehicle state $\mathbf{X}(k)$ using sensor measurements $\mathbf{Z}(k)$, process $\mathbf{N}(k-1)$ and measurement $\mathbf{M}(k)$ noise.

How does it work?



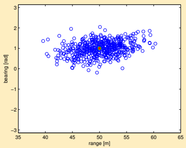
$$\mathbf{X}(k) = f(\mathbf{X}(k-1), \mathbf{N}(k-1)) \; \textit{prediction}$$
$$\mathbf{Z}(k) = h(\mathbf{X}(k), \mathbf{M}(k)) \; \textit{measurement}$$

- "standard" in nonlinear estimation, *Bayesian* approach
- 1^{st} order Taylor expansion of the nonlinear functions
- provides estimation uncertainties in form of error covariance matrices
- difficult tuning
- reliable for systems that are almost linear on the time scale of the updates

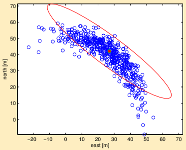
Interesting alternative: Unscented Kalman Filter (UKF)

The goal stays the same: to estimate the state $\mathbf{X}(k)$: mean ($\hat{X}$) and covariance (P).

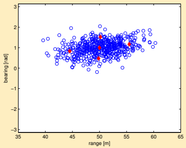
e.g. Mapping from Polar to Cartesian coordinates



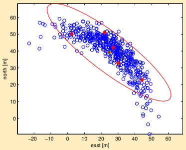
(a) polar coord.



(b) $\hat{X}, P$ (EKF)



(c) UT sampled



(d) $\hat{X}, P$ (UKF)

EKF

linearisation of the transform

UKF

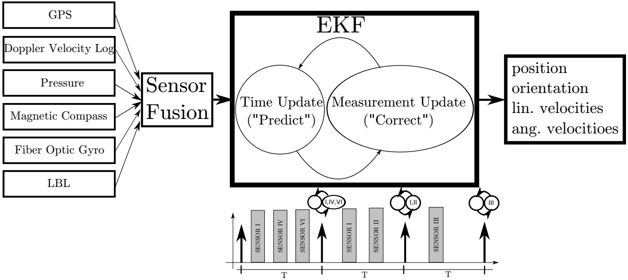
pdf is estimated using a collection of samples of a GRV propagated through the nonlinear transformation

Unscented Transform (UT): selects the samples of a Gaussian and assigns a weight to each.

- guarantees 2^{nd} order Taylor expansion accuracy [2]
- no infamous Jacobians
- same computational cost as EKF

Sensor Fusion

Sensor measurements are not available all the time - messages from sensors arrive at different moments and sensors could be unavailable due to different causes.



Gather the information in between the cycles and integrate it together in the measurement model (Eq. 4).

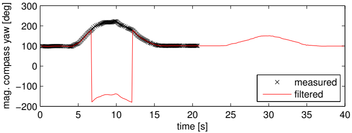
$$\mathbf{Z}(k) = \begin{bmatrix} \mathbf{Z}_{sen.I} \\ \mathbf{Z}_{sen.II} \end{bmatrix}, \mathbf{H}(k) = \begin{bmatrix} \mathbf{H}_{sen.I} \\ \mathbf{H}_{sen.II} \end{bmatrix}, \mathbf{R}(k) = \begin{bmatrix} \mathbf{R}_{sen.I} & 0 \\ 0 & \mathbf{R}_{sen.II} \end{bmatrix} \quad (4)$$

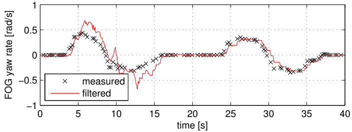
$$\mathbf{Z}(k) = h(\mathbf{X}(k), \mathbf{M}(k)) = \mathbf{H}\mathbf{X}(k \mid k-1) + \mathbf{M}(k)$$

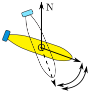
$\mathbf{R}_{sen.I}$ and $\mathbf{R}_{sen.II}$ are defining sensor measurement (un)certainty.

Why Kalman?

I. Blending the measurements





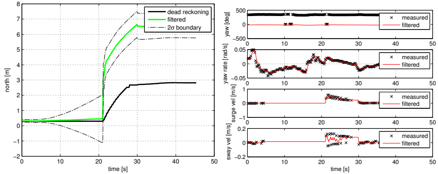


yaw (ψ) is inferred from compass & FOG.

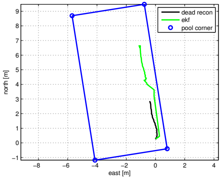
EKF benefit: ✓ if one of them stops working - the other one tries to compensate the failure

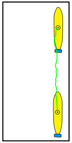
II. Prediction & Filtering

If the measurement temporarily fails, EKF keeps predicting. Noisy velocity information is filtered.



Outcome: ✓ less drift





III. Parametric Estimation: setting the values of model and observation noise covariance matrices dozes the "trust" in model/measurements.

Performing Localisation in Real Missions

Spiral surfacing trajectory

Spiral trajectory and surfacing action was taken with Nessie starting from the depth of around 12 m (Fig. 3).

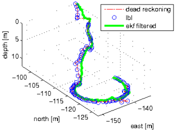
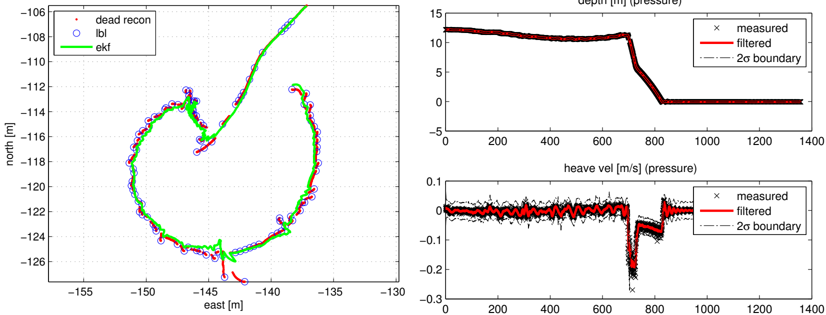


Figure 3: Spiral trajectory localisation



(a) N/E localisation (b) depth filtering

AUV Localisation using EKF in Practise

There is no is no exact ground truth for underwater robot localisation available

Issues to Address in AUV Localisation

Heading Measurement

 vs. 

Compass (*yaw*)

✗ prone to noise

✗ calibration

✓ absolute

Gyro (*yaw rate*)

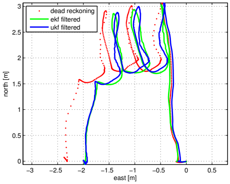
✓ accurate

✗ relative

Apart from being "fused", measurements of *yaw rate* and *yaw* can be taken with adjustable confidence.

Nonlinearity

UKF [2] is an interesting alternative in handling nonlinearities.

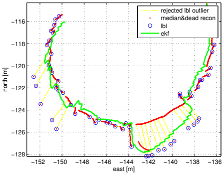


✓ model emulation

✓ computational cost

LBL imprecision

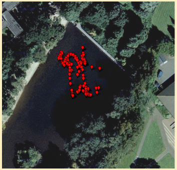
Position updates can deviate from the trajectory, resulting in outliers. EKF was compared with median filter.

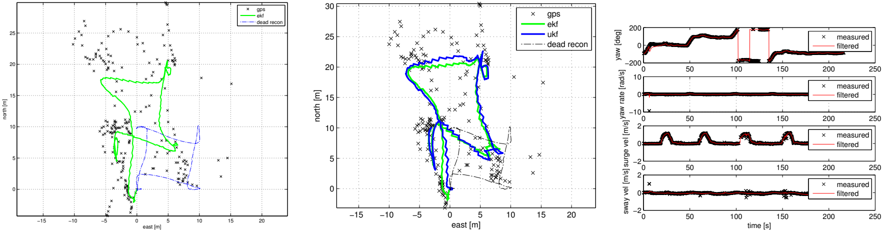


✓ robust

Rectangular trajectory in low depth

Fairly noisy GPS signal (antenna is on the surface) was used as position correction. Parameters $\sigma_{north}, \sigma_{east}$ of north and east measurement noise can be set (4(c), 4(d)). Reasonably chosen values significantly correct the result obtained with pure dead reckoning.





(c) N/E localisation $\sigma = 1.0m$ (d) N/E localisation $\sigma = 0.5m$ (e) Velocities & heading

Figure 4: Rectangular trajectory localisation

Conclusions

Presented work reports the design of an navigation module for a real AUV based on Extended Kalman Filter (EKF) .

- EKF proves to be useful navigation tool with satisfactory navigation performance and several convenient features:
  capable of successfully combining together different sensory information (*sensor fusion*)
  estimate that tends to be optimal with respect to set expectations
  recovering from the missing measurements
  filtering corrupted position information: outliers or signal noise
- 5DOF *constant velocity* mathematical model for state prediction was introduced
- Suitable management of heading measurement was addressed and the role of EKF in correcting deficiencies
- Nonlinearity issues have been investigated with the usage of Unscented Kalman Filter (UKF)

Future Work

- More trials, particularly ones where the vehicle trajectory has been fixed to known landmarks so that the results of localisation could be thoroughly evaluated with trustful ground truth
- Navigation of the tilted vehicle movements in order to make an evaluation of the influence of the 5th d.o.f
- EKF could be improved so that it works with control inputs
- Filtering of the noisy absolute position gives space for improvement. Solution for rejecting outliers could rely on some version of *back-filtering* - filtering based on history of received observations.

References

D. Ribas, P. Ridao, and J. Neira. (2010)
Underwater Slam for Structured Environments using an Imaging Sonar.
Springer Verlag

Julier, S. and Uhlmann, J.K. (1996)
A general method for approximating nonlinear transformations of probability distributions.
Robotics Research Group, Department of Engineering Science, University of Oxford

S. Thrun, W. Burgard, and D. Fox. (2005)
Probabilistic robotics
MIT Press

Grewal, M.S. and Andrews, A.P. (2001)
Kalman filtering: theory and practice using MATLAB.
Wiley Online Library

Ristic, B. and Arulampalam, S. and Gordon, N. (2004)
Beyond the Kalman filter: Particle filters for tracking applications.
Artech House Publishers

The End